

CLAIMS

What is claimed is:

1. A method, comprising:

initializing a debugger in a first computer system during the pre-boot phase of the first computer system, the debugger to operate from a firmware environment of the first computer system;

initializing a communication channel of the first computer system to enable a second computer system to be communicatively coupled to the first computer system;

entering the debugger in response to a debug event; and

examining the first computer system with the debugger.

2. The method of claim 1, further comprising communicatively coupling a second computer system to the first computer system at the communication channel.

3. The method of claim 2 wherein the debug event comprises receiving a debug command from the second computer system at the first computer system.

4. The method of claim 3, further comprising setting a watchdog timer at the first computer system to periodically check for the debug command at the communication channel.
5. The method of claim 2 wherein the debug event comprises receiving a wake-event from the second computer system at the first computer system.
6. The method of claim 5 wherein the wake-event comprises a Wake-on Local Area Network (LAN) packet.
7. The method of claim 5 wherein an operating system executing on the first computer system is hung.
8. The method of claim 1 wherein the debug event comprises detecting an exception that invokes an exception handler installed on the first computer system.
9. The method of claim 1 wherein examining the first computer system comprises:
 - gathering machine state information about the first computer system; and
 - sending the machine state information to the second computer system from the first computer system.

10. The method of claim 1, further comprising communicatively coupling a third computer system to the second computer system, the third computer system to interact with the debugger via the second computer system.

11. The method of claim 1 wherein the debugger is executable by the first computer system independent of an operating system of the first computer system.

12. The method of claim 1 wherein the debugger is operable during the pre-boot phase, an operating system (OS) runtime phase, and an after-life phase of the first computer system.

13. An article of manufacture comprising:

a machine-accessible medium including a plurality of instructions which when executed perform operations comprising:

initializing a debugger in a computer system during the pre-boot phase of the computer system, the debugger to operate from a firmware environment of the computer system;

entering the debugger in response to a debug event; and

examining the computer system with the debugger.

14. The article of manufacture of claim 13 wherein the debug event comprises detecting an exception that invokes an exception handler installed on the computer system.

15. The article of manufacture of claim 13 wherein the debug event comprises receiving a wake-event from a second computer system communicatively coupled to the computer system.

16. The article of manufacture of claim 13 wherein the debug event comprises receiving a debug command from a second computer system communicatively coupled to the computer system.

17. The article of manufacture of claim 13 wherein the debugger to be operable during the pre-boot phase, an operating system (OS) runtime phase, and an after-life phase of the computer system.

18. The article of manufacture of claim 13 wherein the plurality of instructions to be executable independent an operating system executable on the computer system.

19. The article of manufacture of claim 13 wherein the plurality of instructions to operate substantially in compliance with an Extensible Firmware Interface (EFI) specification.

20. A system, comprising:

a first computer system, comprising:

a processor;

a communication channel operatively coupled to the processor; and

at least one flash memory device operatively coupled to the processor,

the at least one flash memory device including firmware instructions which

when executed by the processor perform operations comprising:

initializing a debugger in the first computer system during the pre-boot phase of the first computer system, the debugger to operate from a firmware environment of the first computer system;

initializing the communication channel;

entering the debugger in response to a debug event; and

examining the first computer system with the debugger.

21. The system claim 20 wherein the debug event comprises detecting an exception that invokes an exception handler installed on the first computer system.

22. The system of claim 20 further comprising a second computer system communicatively coupled to the first computer system at the communication channel.

23. The system of claim 22 wherein the debug event comprises receiving a wake-event from the second computer system.

24. The system of claim 22 wherein the debug event comprises receiving a debug command from the second computer system.

25. The system of claim 22 further comprising a third computer system communicatively coupled to the second computer system, the third computer system to interact with the debugger via the second computer system.

26. The system of claim 20 wherein the firmware instructions to be executable independent of an operating system executable on the first computer system.

27. The system of claim 20 wherein the debugger is operable during the pre-boot phase, an operating system (OS) runtime phase, and an after-life phase of the computer system.

28. The system of claim 20 wherein the firmware instructions to operate substantially in compliance with an Extensible Firmware Interface (EFI) specification.